REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED
Reprint

4. TITLE AND SUBTITLE
TITLE ON ~~REPRINT~~

5. FUNDING NUMBERS
DAAG55-98-1-0230

6. AUTHOR(S)
AUTHOR(S) ON ~~REPRINT~~

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
University of Texas - Austin

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 37634.26-PH

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

ABSTRACT ON ~~REPRINT~~

20011101 128

14. SUBJECT TERMS

15. NUMBER IF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Chapter 6

Robust Order Statistics based Ensembles for Distributed Data Mining

Kagan Tumer and Joydeep Ghosh

6.1 Mining of Distributed Data Sources

An implicit assumption in traditional statistical pattern recognition and machine learning algorithms is that the data to be used for model development is available as a single flat file. This assumption is valid for virtually all popular benchmark datasets such as those available from ELENA, Statlog or the UCI machine learning repository. Such datasets are small or medium sized, requiring a few megabytes at most. Thus the algorithms typically also assume that the entire data can fit in main memory, and do not address computational issues regarding scalability and "out-of-core" operations.

The tremendous explosion in the amount of data gathering and warehousing in the past few years has generated very large and complex databases.

Any effort in mining information from such databases has to address the fact that (1) data may be kept in several files as in interlinked relational databases, and information needed for decision making may be spread over more than one file. For example, the concept of "collective data mining" (see chapter 5) explicitly addresses "vertical partitioning" situations where the features or variables relevant to a classification decision are spread over multiple files, each accessible to only one classifier; (2) the files may be spread across several disks or even across different geographical locations, and (3) the statistical quality of data may vary widely. For example the percentage of cases involving financial or health-care fraud varies in different regions, and so does the amount of missing information.

One can argue that by transfering all data to a single warehouse and performing a series of merges and joins, one can get a single (albeit very large), flat file. A traditional algorithm can be used after randomizing and subsampling this file. But in real applications this approach may not be feasible because of the computational, bandwidth and storage costs. In certain cases, it may not even be possible for a variety of practical reasons including security, privacy, proprietary nature of data, need for fault tolerant distribution of data and services, real-time processing requirements, statutary constraints imposed by law, etc. (see chapter 3). Then there are two options. If the owners of the individual databases are willing to provide high level or summary information/decisions such as local classification estimates, and transmit this information to a central location, then a metalearner can be applied to the component decisions to come up with a final, composite decision. Note that such high level information not only has reduced storage and bandwidth requirements, but also maintains the privacy of individual records (DuMouchel 1999. Otherwise one has to resort to a distributed computing framework such as the emerging field of collective intelligence (COIN), wherein techniques are developed such that local and independent computations can still increase a desired global utility function (Wolpert and Tumer 1999).

The first option leads to several issues reminiscent of studies in decision fusion (Dasarathy 1994) applied largely to multi-sensor fusion and distributed control problems. It is also related to the theory of ensembles (Sharkey 1996) and integration of multiple learned models (Chan, Stolfo, and Wolpert 1996). But there are substantial new aspects that need to be addressed in a distributed data mining context.

This chapter is rooted in the ensemble framework and shows how order statistics can be used in the design of a "metalearner" that examines the outputs of multiple distributed classifers and provides a final decision. Order statistics is one of the key tools of robust statistics, tailored to handling data with outliers. In a distributed data mining scenario in which there is wide variability among the individual classifers because of the underlying quality of the local data that they examine, a metalearner should be able to tolerate a few outlier classifier results. The robust properties of order statistics based approaches such as median filtering and m-estimators (Arnold, Balakrishnan, and Nagaraja 1992), have been observed in many disciplines. Thus they are an obvious candidate for metalearning in such environments.

The next section provides a brief review of the metalearning framework for classification to put the proposed techniques in perspective. Section 6.3 summarizes the relationship between classifier errors and decision boundaries and provides the necessary background for mathematically analyzing order statistic combiners (Tumer and Ghosh 1996a). Section 6.4 introduces simple order statistic combiners. Based on these concepts, in section 6.5 we propose two powerful combiners, *trim* and *spread*, and derive the amount of error reduction associated with each. In section 6.6 we present the performance of order statistic combiners on several datasets. Section 6.7 discusses the implications of using linear combinations of order statistics as a strategy for pooling the outputs of individual classifiers.

6.2 A Brief History of Multi-Learner Systems

The idea of integrating multiple models for the *same* problem, has been examined for a long time. The main goal is to obtain a better composite global model, with more accurate and reliable estimates or decisions. Some notable early systems include Selfridge's Pandemonium (Selfridge 1958) where a head-deamon would select the deamon that "shouted the loudest," and Nilsson's committee machines. A strong motivation for such systems was voiced by Kanal in his classic 1974 paper (Kanal 1974):

> "It is now recognized that the key to pattern recognition problems does not lie wholly in learning machines, statistical approaches, spatial, filtering, ..., or in any other particular solution which has been vigorously advocated by one or another group during the last one and a half decades as the solution to the pattern recognition problem. No single model exists for all pattern recognition problems and no single technique is applicable to all problems. Rather what we have is a bag of tools and a bag of problems."

In the late 1970s, much work was done on combining linguistic and statistical models, and on combining heuristic search with statistical pattern recognition. Subsequently, similar sentiments on the importance of multiple approaches were also voiced in the AI community (Minsky 1991):

> "To solve really hard problems, we'll have to use several different representations. ... It is time to stop arguing over which type

> of pattern-classification technique is best. ... Instead we should work at a higher level of organization and discover how to build managerial systems to exploit the different virtues and evade the different limitations of each of these ways of comparing things."

Integration of multiple data sources and / or learned models can now be found in several disciplines, for example, the combining of estimators in econometrics (Granger 1989), evidences in rule-based systems (Barnett 1981) and multi-sensor data fusion (Dasarathy 1994). Of course, one can find numerous examples in the human central nervous system (Shepherd 1979), as well as in some large engineering systems such as those that demand fault tolerance or employing control mechanisms that may need to function in different operating regimes (Narendra, Balakrishnan, and Ciliz 1995). In particular, multiple models for nonlinear control has a long tradition.[1] Hybridization in a broader sense is seen in efforts to combine two or more of neural network, Bayesian, GA, fuzzy logic and knowledge-based systems (Aggarwal, Ghosh, Nair, and Taha 1996,Taha and Ghosh 1997). The goal is again to incorporate diverse sources and forms of information and to exploit the somewhat complementary nature of different methodologies.

Figure 6.1 shows a generic diagram of an *ensemble*, the simplest and most well understood type of multi-learner systems. Each component learner is a regressor or classifier, trying to solve the same task. While data ultimately originates from an underlying universal set X, each learner may receive somewhat different subsets of the data for "training" or parameter estimation (as in bagging [Breiman, 1994] and boosting [Drucker, Cortes, Jackel, LeCun, and Vapnik 1994]), and may be using different feature extractors (fs) on the same raw data. For example, in our earlier work on sonar classification (Ghosh, Deuser, and Beck 1992), Fourier, wavelet and autoregressive coefficients extracted from the same preprocessed time series were used respectively for three different classifier types.

Along with selection of training samples and feature extractors, one needs to decide what types of learners to use and how many, and finally, how to design the metalearner. There are also larger issues of how to train the components given that they are part of a bigger system, and to estimate the overall gains achievable.

The simplest metalearner is the *combiner*, where the output y is determined solely from the outputs of the individual learners. In the past few years, a host of experimental results from both neural network and machine learning communities show that combining the outputs of multiple regressors or classifiers via (weighted) averaging, majority vote, product rule, entropy, etc., pro-

[1] See www.itk.ntnu.no/ansatte/Johansen_ Tor.Arne/mmamc/address.html for a detailed list of researchers in this area.

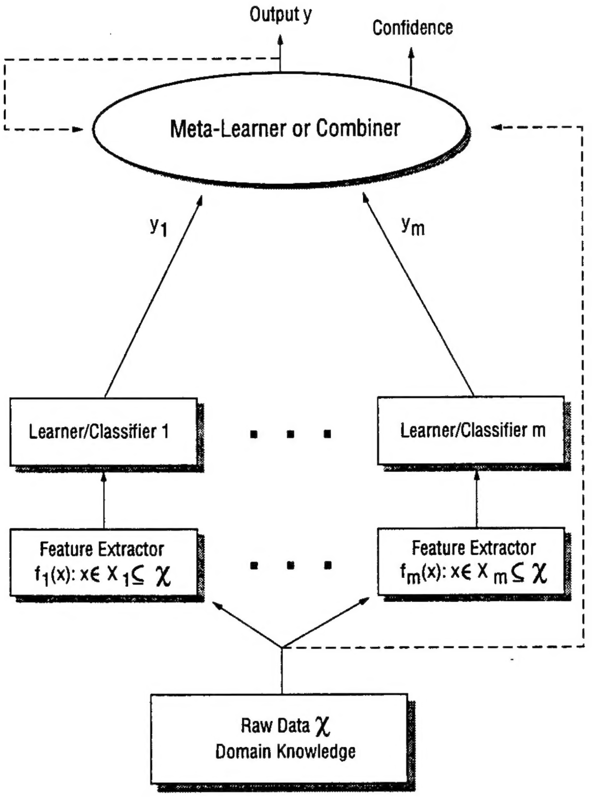

Figure 6.1: Generic architecture of a multi-learner system for regression or classification

vides statistically significant improvement in performance along with tighter confidence intervals. Moreover, theoretical analysis has been developed for both regression (Perrone 1993, Hashem and Schmeiser 1993) and classification (Tumer and Ghosh 1996a, 1999), to estimate the gains achievable.

Beyond simple combiners are metalearning methods such as arbitration (Chan and Stolfo 1997) and stacking (Wolpert 1992). Also in this category are divide-and-conquer approaches, where relatively simple learners specialize in different parts of the input-output space, and the total model is a (possibly soft) union of such simpler models. Techniques of modular learning include "mixtures-of-experts," local linear regression, CART/MARS, adaptive subspace models, etc. (Jordan and Jacobs 1994; Ramamurti and Ghosh 1999; Holmstrom, Koistinen, Laaksonen, and Oja 1997). What distinguishes all these models from simple combiners is that the metalearner's actions now also depends on the current input and/or the target values, i.e. the dotted lines in figure 6.1 are also used.

The simple combining methods are best suited for problems where the individual classifiers perform the same task, and have comparable success. However, such combiners are more susceptible to outliers and to unevenly performing classifiers. On the other hand, the more general metalearners are conceptually more powerful but are vulnerable to all the problems associated with the added learning (e.g., overparameterizing, lengthy training time). They may also require access to the entire input database which may be impractical in a distributed data mining environment.

The performance of classifier ensembles has been spectacular. Brieman calls the combination of decision tree classifiers with boosting the most significant development in classifier design in this decade. Indeed, ensemble methods are even becoming available in commercial data mining tools such as SAS Enterprise Miner Version 3.0. To see intuitively why ensembles have been so effective, note that different types of classifiers have different "inductive biases" (Geman, Bienenstock, and Doursat 1992; Mitchell 1997), and thus, in general, they do not generalize in identical ways even when they are trained on the same data set, and have comparable performance on a test set. Traditionally, the classifier perceived as the "best," as indicated by a suitable scoring function such as a cross-validation based estimation of the true generalization error, is selected. This means that the other classifiers that had been designed during the model exploration/development phase, get discarded. But this results in a potential loss of useful information and effort. An ensemble can effectively make use of such complementary information to reduce model variance (Perrone 1993, Tumer and Ghosh 1999) and in certain situations it also reduces bias as shown by the theory of large margin classifiers (Bartlett and Shawe-Taylor, 1998). It works best when each learner is well trained, but different learners generalize in different ways, i.e., there is diversity in the ensemble (Krogh and Vedelsby 1995). Diversity may be induced through different presentations of the input data, as in bagging, variations in learner design, or by adding a penalty to the ouputs to encourage diversity.

For large data sets, there is also a computational reason for using metalearners, namely, it can make certain inductive learners fairly scalable to large data sets (Chan and Stolfo 1997; Provost and Kolluri 1999; also see chapter 1 in this book). In Chan and Stolfo (1997) each component classifier uses a randomly chosen subset of the entire data set, and the decision tree classifiers can operate in parallel. This proves quite effective as overall computation can be greatly reduced with little loss in performance. A beneficial side-effect of using smaller datasets for decision tree classifiers is that the classifiers obtained are smaller in size (Oates and Jensen 1998). These two as well as other scalability techniques for decision tree based classifiers are covered in a nice recent survey (Provost and Kolluri 1999).

A fundamental assumption in all the multi-classifier approaches mentioned above is that the designer has access to the entire data set, which can be used in its entirety, resampled in a random (bagging) or weighted (boosting) way (Breiman 1999), or randomly partitioned and distributed. Thus, except for boosting situations, each classifier sees training data of comparable quality. If the individual classifiers are then appropriately chosen and trained properly, their performances will be (relatively) comparable in any region of the problem space. So gains from combining are derived from the diversity among classifiers rather that by compensating for weak members of the pool.

This assumption is clearly invalid for distributed data mining using heterogenous sites (see chapter 5). Such real-life conditions often result in a pool of classifiers that may have significant variations in their overall performance. Moreover, they may lead to conditions where individual classifiers have similar average performance, but substantially different performance over different parts of the input space. In such cases, combining is still desirable, but neither simple combiners nor metalearners are particularly well-suited for the type of problems that arise. For example, the simplicity of averaging the classifier outputs is appealing, but the prospect of one poor classifier corrupting the combiner makes this a risky choice. Weighted averaging of classifier outputs appears to provide some flexibility (Hashem and Schmeiser 1993, Merz and Pazzani 1997). Unfortunately, the weights are still assigned on a per classifier basis rather than a per sample or per class basis. If a classifier is accurate only in certain areas of the input space, this scheme fails to take advantage of the variable accuracy of the classifier in question. Using a metalearner that provides different weights for different patterns can potentially solve this problem, but at a considerable cost. Also, as explained earlier, the off-line training of a metalearner using substantial amount of data outputted by geographically distributed classifiers, may not be feasible or even allowable. In addition to providing robustness demanded in such situations, the order statistic combiners presented in this work also aim at bridging the gap between simplicity and generality by allowing the flexible selection of classifiers without the associated cost of training metaclassifiers.

6.3 Error Characterization in a Single Classifier

In this section we summarize the approach and results of Tumer and Ghosh (1996a, 1999),[2] that quantify the effect of inaccuracies in estimating a posterior class probabilities on the classification error for a single classifier. This

[2] This and other related papers can be downloaded from www.lans.ece.utexas.edu.

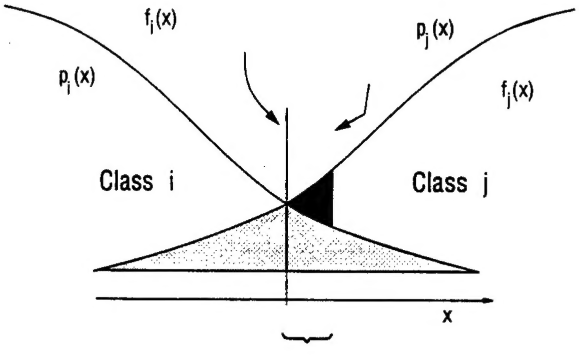

Figure 6.2: Error regions associated with approximating the a posteriori probabilities (Tugh 1996).

background is needed to characterize and understand the impact of order statistics combiners, as described in sections 6.3 and 6.4.

It is well known that, given *one-of-L* desired outputs and sufficient training samples reflecting the class priors, the outputs of certain classifiers trained to minimize a mean square or cross-entropy error criteria, approximate the a posteriori probability densities of the corresponding classes (Richard and Lippmann 1991; Ruck, Rogers, Kabrisky, Oxley, and Suter 1990). Based on this result, one can model the ith output of the mth such classifier as:

$$f_i^m(x) = p_i(x) + \epsilon_i^m(x), \tag{6.1}$$

where $p_i(x)$ is the true posterior for ith class on input x, and $\epsilon_i^m(x)$ is the error of the mth classifier in estimating that posterior.

Now, let us decompose the error into two parts: $\epsilon_i^m(x) = \beta_i^m + \eta_i^m(x)$. The first component does not vary with the input, and provides an offset, or systematic error for each class. The second component gives the variability from that systematic error, for each x in each class, and has zero mean and variance $\sigma^2_{\eta_i^m(x)}$. These two components of the error are similar to the bias and variance decomposition for a quadratic loss function given in Geman, Bienenstock, and Doursat (1992), although they are at the individual input level. We will therefore refer to classifiers as "biased" and "unbiased" implying $\beta_k^m \neq 0$ for some k, m, and $\beta_k^m = 0, \forall k, m$, respectively. Let b^m denote the offset between the ideal class boundary, x^* (based on $p_i(x) = p_j(x)$) and the realized boundary, x_b^m (based on $f_i^m(x) = f_j^m(x)$), as shown in figure 6.2 (Tumer and Ghosh 1996a). This boundary offset ($b^m = x_b^m - x^*$) has mean and variance given respectively by:

$$\beta^m = \frac{\beta_i^m - \beta_j^m}{s}, \tag{6.2}$$

and

$$\sigma_{b^m}^2 = \frac{\sigma_{\eta_i^m(x)}^2 + \sigma_{\eta_j^m(x)}^2}{s^2}, \tag{6.3}$$

where $s = p_j'(x^*) - p_i'(x^*)$ as introduced in Tumer and Ghosh (1996a).

Let us further denote the probability density function of this boundary offset by $f_b(x)$. The expected model error associated with the selection of a particular classifier m, can then be expressed as:

$$E_{model}^m = \int_{-\infty}^{\infty} A(b) f_b(b) db, \tag{6.4}$$

where $A(b) = \int_{x^*}^{x^*+b} \left(p_j(x) - p_i(x)\right) dx$ is the error due to the selection of a particular decision boundary. In general, it is not possible to obtain the density function for the boundary offset without making assumptions on the distributions of the errors. However, a first order approximation, derived in Tumer and Ghosh (1996a), leads to:

$$E_{model}^m = \int_{-\infty}^{\infty} \frac{1}{2} b^2 s f_b(b) db. \tag{6.5}$$

Let us define the first and second moments of the boundary offset as follows:

$$\mathcal{M}_1 = \int_{-\infty}^{\infty} x f_b(x) dx and \mathcal{M}_2 = \int_{-\infty}^{\infty} x^2 f_b(x) dx.$$

If the individual classifiers are unbiased, the offset b^m of a single classifier has $\mathcal{M}_1 = 0$ and $\mathcal{M}_2 = \sigma_{b^m}^2$, leading to:

$$E_{model}^m = \frac{s\mathcal{M}_2}{2} = \frac{s\sigma_{b^m}^2}{2}. \tag{6.6}$$

Now, if the classifiers are biased, the variance of b is left unchanged (given by Equation 6.3), but the mean becomes $\beta = \frac{\beta_i - \beta_j}{s}$. In other words, we have $\mathcal{M}_1 = \beta^m$ and $\sigma_{b^m}^2 = \mathcal{M}_2 - \mathcal{M}_1{}^2$, leading to the following model error:

$$E_{model}^m(\beta) = \frac{s\mathcal{M}_2}{2} = \frac{s}{2}(\sigma_{b^m}^2 + (\beta^m)^2). \tag{6.7}$$

To emphasize the distinction between biased and unbiased classifiers, the model error will be given as a function of β for biased classifiers. A more detailed derivation of class boundaries and error regions is presented in Tumer and Ghosh (1996a). For analyzing the error regions after combining and comparing them to the single classifier case, one needs to determine how the first and second moments of the boundary distributions are affected by combining. The following sections focus on obtaining those values for order statistics based combiners.

6.4 Combining Multiple Classifiers through Order Statistics

AUTHOR: PLEASE INSERT AT LEAST TWO SENTENCES BETWEEN THE HEADING AND SUBHEADING. HEADINGS CANNOT DIRECTLY FOLLOW ONE ANOTHER.

6.4.1 Basic Concepts

In this section, we briefly discuss some basic concepts and properties of order statistics. Let X be a random variable with probability density function $f_X(\cdot)$, and cumulative distribution function $F_X(\cdot)$. Let $(X_1, X_2, \cdots, X_N)$ be a random sample drawn from this distribution. Now, let us arrange them in nondecreasing order, providing:

$$X_{1:N} \leq X_{2:N} \leq \cdots \leq X_{N:N}.$$

The ith order statistic denoted by $X_{i:N}$, is the ith value in this progression. The cumulative distribution function for the smallest and largest order statistic can be obtained by noting that:

$$F_{X_{N:N}}(x) = P(X_{N:N} \leq x) = \Pi_{i=1}^{N} P(X_{i:N} \leq x) = [F_X(x)]^N$$

and:

$$F_{X_{1:N}}(x) = P(X_{1:N} \leq x) = 1 - P(X_{1:N} \geq x)$$

$$= 1 - \Pi_{i=1}^{N} P(X_{i:N} \geq x)$$

$$= 1 - (1 - \Pi_{i=1}^{N} P(X_{i:N} \leq x) = 1 - [1 - F_X(x)]^N$$

The corresponding probability density functions can be obtained from these equations. In general, for the ith order statistic, the cumulative distribution function gives the probability that exactly i of the chosen X's are less than or equal to x. The probability density function of $X_{i:N}$ is then given by David (1970):

$$f_{X_{i:N}}(x) = \frac{N!}{(i-1)!(N-i)!} \left[F_X(x)\right]^{i-1} \left[1 - F_X(x)\right]^{N-i} f_X(x). \quad (6.8)$$

This general form however, cannot always be computed in closed form. Therefore, obtaining the expected value of a function of x using equation 6.8 is not always possible. However, the first two moments of the density function are widely available for a variety of distributions (Arnold, Balakrishnan, and Nagaraja 1992). These moments can be used to compute the expected values of certain specific functions, e.g., polynomials of order less than two.

6.4.2 Combining Unbiased Classifiers through Order Statistics

Now, let us turn our attention to order statistics (OS) combiners. For a given input x, let the network outputs of each of the N classifiers for each class i be ordered in the following manner:

$$f_i^{1:N}(x) \leq f_i^{2:N}(x) \leq \cdots \leq f_i^{N:N}(x).$$

Then one constructs the kth order statistic combiner, by selecting the kth ranked output for each class ($f_i^{k:N}(x)$), as representing its posterior.

In particular, max, med and min combiners are defined as follows:

$$f_i^{max}(x) = f_i^{N:N}(x), \quad (6.9)$$

$$f_i^{med}(x) = \begin{cases} \frac{f_i^{\frac{N}{2}:N}(x) + f_i^{\frac{N}{2}+1:N}(x)}{2} & \text{if } N \text{ is even} \\ f_i^{\frac{N+1}{2}:N}(x) & \text{if } N \text{ is odd,} \end{cases} \quad (6.10)$$

$$f_i^{min}(x) = f_i^{1:N}(x). \quad (6.11)$$

These three combiners are relevant because they represent important qualitative interpretations of the output space. Selecting the maximum combiner is equivalent to selecting the class with the highest posterior. Indeed, since the network outputs approximate the class a posteriori distributions, selecting the maximum reduces to selecting the classifier that is the most "certain" of its decision. The drawback of this method however is that it can be compromised by a single classifier that repeatedly provides high values. The selection of the minimum combiner follows a similar logic, but focuses on classes that are unlikely to be correct, rather than on the correct class. Thus, this combiner eliminates less likely classes by basing the decision on the lowest value for a given class. This combiner suffers from the same ills as the *max* combiner. However, it is less dependent on a single error, since it performs a min-max operation, rather than a max-max[3]. The median classifier on the other hand considers the most "typical" representation of each class. For highly noisy data, this combiner is more desirable than either the min or max combiners since the decision is not compromised as much by a single large error.

The analysis that follows does not depend on the particular order statistic chosen. Therefore, we will denote all OS combiners by $f_k^{os}(x)$ and derive the model error, E_{model}^{os}. The network output provided by $f_k^{os}(x)$ is given by:

$$f_k^{os}(x) = p_k(x) + \epsilon_k^{os}(x), \tag{6.12}$$

Let us first investigate the zero-bias case ($\beta_k = 0, \forall k$), where we get $\epsilon_k^{os}(x) = \eta_k^{os}(x)$. Proceeding as in section 6.3, the boundary b^{os} is shown to be:

$$b^{os} = \frac{\eta_i^{os}(x_b) - \eta_j^{os}(x_b)}{s}. \tag{6.13}$$

For *i.i.d.* η_k's, the first two moments will be identical for each class. Moreover, taking the order statistic will shift the mean of both η_i^{os} and η_j^{os} by the same amount, leaving the mean of the difference unaffected. Therefore, b^{os} will have zero mean, and variance:

$$\sigma_{b^{os}}^2 = \frac{2\sigma_{\eta_k^{os}}^2}{s^2} = \frac{2\alpha\sigma_{\eta_k^m}^2}{s^2} = \alpha\sigma_{b^m}^2, \tag{6.14}$$

where α is a reduction factor that depends on the order statistic and on the distribution of b. For most distributions, α can be found in tabulated form (Arnold, Balakrishnan, and Nagaraja 1992). For example, table 6.1 provides α values for all order statistic combiners, up to 10 classifiers, for a Gaussian distribution (Arnold, Balakrishnan, and Nagaraja 1992, Sarhan and Greenberg 1956). (Because this distribution is symmetric, the α values of l and k where $l + k = N + 1$ are identical, and listed in parenthesis).

Returning to the error calculation, we have: $\mathcal{M}_1^{os} = 0$, and $\mathcal{M}_2^{os} = \sigma_{b^{os}}^2$, providing:

$$E_{model}^{os} = \frac{s\mathcal{M}_2^{os}}{2} = \frac{s\sigma_{b^{os}}^2}{2} = \frac{s\alpha\sigma_{b^m}^2}{2} = \alpha E_{model}^{m}. \tag{6.15}$$

Equation 6.15 shows that the reduction in the error due to using the OS combiner instead of the mth classifier is directly related to the reduction in the variance of the boundary offset b. Since the means and variances of order statistics for a variety of distributions are widely available in tabular form, the reductions can be readily quantified.

[3] Recall that the pattern is ultimately assigned to the class with the highest combined output.

N	k	α	N	k	α	N	k	α
1	1	1.00	6	2 (5)	.280	9	1 (9)	.357
2	1 (2)	.682		3 (4)	.246		2 (8)	.226
3	1 (3)	.560	7	1 (7)	.392		3 (7)	.186
	2	.449		2 (6)	.257		4 (6)	.171
4	1 (4)	.492		3 (5)	.220		5	.166
	2 (3)	.360		4	.210	10	1 (10)	.344
5	1 (5)	.448	8	1 (8)	.373		2 (9)	.215
	2 (4)	.312		2 (7)	.239		3 (8)	.175
	3	.287		3 (6)	.201		4 (7)	.158
6	1 (6)	.416		4 (5)	.187		5 (6)	.151

Table 6.1: Reduction factors α for the Gaussian distribution, based on Sarhan and Greenberg (1956).

6.4.3 Combining Biased Classifiers through Order Statistics

In this section, we analyze the error regions for biased classifiers. Let us return our attention to b^{os}. First, note that the error terms can no longer be studied separately, since in general $(a+b)^{os} \neq a^{os} + b^{os}$. We will therefore need to specify the mean and variance of the result of each operation[4]. Equation 6.13 becomes:

$$b^{os} = \frac{(\beta_i + \eta_i(x_b))^{os} - (\beta_j + \eta_j(x_b))^{os}}{s}. \tag{6.16}$$

Let $\bar{\beta}_k = \frac{1}{N}\sum_{m=1}^{N} \beta_k^m$ be the mean of classifier biases. Since η_k^m's have zero-mean, $\beta_k + \eta_k(x_b)$ has first moment $\bar{\beta}_k$ and variance $\sigma^2_{\eta_k^m} + \sigma^2_{\beta_k^m}$, with $\sigma^2_{\beta_k^m} = E[(\beta_k^m)^2] - \bar{\beta}_k^{\,2}$, where $[\cdot]$ denotes the expected value operator.

Taking a specific order statistic of this expression will modify both moments. The first moment is given by $\bar{\beta}_k + \mu^{os}$, where μ^{os} is a shift which depends on the order statistic chosen, but not on the class. Then, the first moment of b^{os} is given by:

$$\frac{(\bar{\beta}_i + \mu^{os}) - (\bar{\beta}_j + \mu^{os})}{s} = \frac{\bar{\beta}_i - \bar{\beta}_j}{s} = \bar{\beta}. \tag{6.17}$$

Note that the bias term represents an "average bias" since the contributions due to the order statistic are removed. Therefore, reductions in bias cannot be obtained from a table similar to table 6.1.

Now, let us turn our attention to the variance. Since $\beta_k^m + \eta_k^m(x_b)$ has variance $\sigma^2_{\eta_k^m} + \sigma^2_{\beta_k^m}$, it follows that $(\beta_k + \eta_k(x_b))^{os}$ has variance $\sigma^2_{\eta_k^{os}} = \alpha(\sigma^2_{\eta_k^m} +$

[4] Since the exact distribution parameters of b^{os} are not known, we use the sample mean and the sample variance.

$\sigma^2_{\beta^m_k}$), where α is the factor discussed in section 6.4.2. Therefore, the variance of b^{os} is given by:

$$\begin{aligned}
\sigma^2_{b^{os}} &= \frac{\sigma^2_{\eta^{os}_i} + \sigma^2_{\eta^{os}_j}}{s^2} = \frac{2\alpha\sigma^2_{\eta^m_i}}{s^2} + \frac{\alpha(\sigma^2_{\beta^m_i} + \sigma^2_{\beta^m_j})}{s^2} \\
&= \alpha(\sigma^2_{b^m} + \sigma^2_{\beta^m}), \qquad (6.18)
\end{aligned}$$

where $\sigma^2_{\beta^m} = \frac{\sigma^2_{\beta^m_i} + \sigma^2_{\beta^m_j}}{s^2}$ is the variance introduced by the systematic errors of different classifiers.

We have now obtained the first and second moments of b^{os}, and can compute the model error. Namely, we have $\mathcal{M}^{os}_1 = \bar{\beta}$ and $\sigma^2_{b^{os}} = \mathcal{M}^{os}_2 - (\mathcal{M}^{os}_1)^2$, leading to:

$$\begin{aligned}
E^{os}_{model}(\beta) &= \frac{s}{2}\mathcal{M}^{os}_2 = \frac{s}{2}(\sigma^2_{b^{os}} + \bar{\beta}^2) \qquad (6.19) \\
&= \frac{s}{2}(\alpha(\sigma^2_{b^m} + \sigma^2_{\beta^m}) + \bar{\beta}^2). \qquad (6.20)
\end{aligned}$$

The reduction in the error is more difficult to assess in this case. By writing the error as:

$$E^{os}_{model}(\beta) = \alpha\frac{s}{2}(\sigma^2_b + (\beta^m)^2) + \frac{s}{2}(\alpha\sigma^2_\beta + \bar{\beta}^2 - \alpha(\beta^m)^2),$$

we get:

$$E^{os}_{model}(\beta) = \alpha E^m_{model}(\beta) + \frac{s}{2}(\alpha\sigma^2_\beta + \bar{\beta}^2 - \alpha(\beta^m)^2). \qquad (6.21)$$

Analyzing the error reduction in the general case requires knowledge about the bias introduced by each classifier. Unlike regression problems where the bias and variance contributions to the error are additive and well-understood, in classification problems their interaction is more complex (Friedman 1997). Indeed it has been observed that ensemble methods do more than simply reduce the variance (Schapire, Freund, and Bartlett 1997).

Based on these observations and equation 6.21, let us analyze extreme cases. For example, if each classifier has the same bias, σ^2_β is reduced to zero and $\bar{\beta} = \beta^m$. In this case the error reduction can be expressed as:

$$E^{os}_{model}(\beta) = \frac{s}{2}(\alpha\sigma^2_b + (\beta^m)^2 = \alpha E^m_{model}(\beta) + \frac{s(1-\alpha)}{2}(\beta^m)^2,$$

where α balances the two contributions to the error. A small value for α will reduce the first component of the error (mainly variance), while leaving the second term untouched. The net effect will be very similar to results obtained for regression problems. In this case, it is important to reduce classifier bias before combining (e.g., by using an overparametrized model).

If on the other hand, the biases produce a zero mean variable, we obtain $\bar{\beta} = 0$. In this case, the model error becomes:

$$E^{os}_{model}(\beta) = \alpha E^{m}_{model}(\beta) + \frac{s\alpha}{2}(\sigma^2_{\beta^m} - (\beta^m)^2)$$

and the error reduction will be significant if the second term is small or negative. In fact, if the variation among the biases is small relative to their magnitude, the error will be reduced more than in the unbiased cases. If however, the variation is large compared to the magnitude, the error reduction will be minimal. Furthermore, if α is large and the biases are small and highly varied, it is possible for this combiner to do worse than the individual classifiers, which is a danger not present for regression problems. This observation very closely parallels results reported in Friedman (1997).

6.5 Linear Combining of Ordered Classifier Outputs

In the previous section, we derived error reductions when the class posteriors are directly estimated through the ordered classifier outputs. Since simple averaging has also been shown to provide benefits, in this section, we investigate the combinations of averaging and order statistics for pooling classifier outputs.

6.5.1 Spread Combiner

The first linear combination of ordered classifier outputs we study focuses on extrema. As discussed in section 6.4.2, the maximum and minimum of a set of classifier outputs carry specific meanings. Indeed, the maximum can be viewed as the class for which there is the most evidence. Similarly, the minimum deletes classes with little evidence. In order to avoid a single classifier from having too large of an impact on the eventual output, these two values can be averaged to yield the *spread* combiner. This combiner strikes a balance between the positive and negative evidence, leading to a more robust combiner than either of them.

Spread Combiner for Unbiased Classifiers:

For a classifier without bias, the spread combiner is formally defined as:

$$f_i^{spr}(x) = \frac{1}{2}(f_i^{1:N}(x) + f_i^{N:N}(x)) = p(c_i|x) + \eta_i^{spr}(x), \qquad (6.22)$$

where:

$$\eta_i^{spr}(x) = \frac{1}{2}\left(\eta_i^{1:N}(x) + \eta_i^{N:N}(x)\right).$$

The variance of $\eta_i^{spr}(x)$ is given by:

$$\begin{aligned}\sigma^2_{\eta_i^{spr}} & \\ &= \tfrac{1}{4}\sigma^2_{\eta_i^{1:N}(x)} + \tfrac{1}{4}\sigma^2_{\eta_i^{N:N}(x)} + \tfrac{1}{2}cov(\eta_i^{1:N}(x), \eta_i^{N:N}(x)).\end{aligned} \quad (6.23)$$

where $cov(\cdot,\cdot)$ represents the covariance between two variables (even when the η_i's are independent, ordering introduces correlations). Note that because of the ordering, the variances in the first two terms of equation 6.23 can be expressed in terms of the individual classifier variances. Furthermore, the covariance between two order statistics can also be determined in tabulated form for given distributions. Table 6.2 provides these values for a Gaussian distribution based on Sarhan and Greenberg (1956). This expression can be further simplified for symmetric distributions where $\sigma^2_{\eta^{1:N}} = \sigma^2_{\eta^{N:N}}$ (e.g., Gaussian noise model) and leads to:

$$\sigma^2_{\eta_i^{spr}} = \frac{1}{2}\left(\alpha_{1:N} + B_{1,N:N}\right)\sigma^2_{\eta_i(x)}, \quad (6.24)$$

where $\alpha_{m:N}$ is the variance of the mth ordered sample and $B_{m,l:N}$ is the covariance between the mth and lth ordered samples, given that the initial samples had unit variance (Sarhan and Greenberg 1956). Because this is a symmetric distribution, the β values are also symmetric (e.g.;, $\beta_{1,2:5} = \beta_{4,5:5}$).

Then, using equation 6.3, the variance of the boundary offset b^{spr} can be calculated:

$$\begin{aligned}\sigma^2_{b^{spr}} &= \frac{\sigma^2_{\eta_i^{spr}} + \sigma^2_{\eta_j^{spr}}}{s^2} \\ &= \frac{1}{2}\left(\alpha_{1:N} + B_{1,N:N}\right)\sigma^2_b.\end{aligned} \quad (6.25)$$

Finally, through equation 6.6, we can obtain the reduction in the model error due to the spread combiner:

$$\frac{E^{spr}_{model}}{E_{model}} = \frac{\alpha_{1:N} + B_{1,N:N}}{2}. \quad (6.26)$$

Based on equation 6.26 and tables 6.1 and 6.2, table 6.3 displays the error reductions provided by the spread combiner for a Gaussian noise model (for comparison purposes, the error reduction for the *min* and *max* combiners is also provided. Note that for the Gaussian distribution, the error reduction of *min* is equal to that of *max*.).

N	k,l	B	N	k,l	B	N	k,l	B	N	k,l	B
2	1,2	.318	6	2,3	.189	8	1,4	.095	9	1,6	.059
3	1,2	.276		2,4	.140		1,5	.075		1,7	.049
	1,3	.165		2,5	.106		1,6	.060		1,8	.040
4	1,2	.246		3,4	.183		1,7	.048		1,9	.031
	1,3	.158	7	1,2	.196		1,8	.037		2,3	.154
	1,4	.105		1,3	.132		2,3	.163		2,4	.117
	2,3	.236		1,4	.099		2,4	.123		2,5	.093
5	1,2	.224		1,5	.077		2,5	.098		2,6	.077
	1,3	.148		1,6	.060		2,6	.079		2,7	.063
	1,4	.106		1,7	.045		2,7	.063		2,8	.052
	1,5	.074		2,3	.175		3,4	.152		3,4	.142
	2,3	.208		2,4	.131		3,5	.121		3,5	.114
	2,4	.150		2,5	.102		3,6	.098		3,6	.093
6	1,2	.209		2,6	.080		4,5	.149		3,7	.077
	1,3	.139		3,4	.166	9	1,2	.178		4,5	.137
	1,4	.102		3,5	.130		1,3	.121		4,6	.113
	1,5	.077	8	1,2	.186		1,4	.091			
	1,6	.056		1,3	.126		1,5	.073			

Table 6.2: Some Reduction Factors B for the Gaussian Distribution, based on Sarhan and Greenberg (1956).

Spread Combiner for Biased Classifiers:

Now, if the classifier biases are nonzero, the spread combiner's output is given by:

$$\begin{aligned} f_i^{spr}(x) &= \frac{1}{2}(f_i^{1:N}(x) + f_i^{N:N}(x)) \\ &= p(c_i|x) + (\eta_i(x) + \beta_i)^{spr}. \end{aligned} \tag{6.27}$$

In that case, the boundary offset is given by:

$$b^{spr} = \frac{(\beta_i + \eta_i(x_b))^{spr} - (\beta_j + \eta_j(x_b))^{spr}}{s}, \tag{6.28}$$

which after expanding each term and regrouping can be expressed as:

$$\begin{aligned} b^{spr} &= \frac{(\beta_i + \eta_i(x_b))^{1:N} - (\beta_j + \eta_j(x_b))^{1:N}}{2s} \\ &\quad + \frac{(\beta_i + \eta_i(x_b))^{N:N} - (\beta_j + \eta_j(x_b))^{N:N}}{2s}. \end{aligned} \tag{6.29}$$

The first moment of b^{spr} can be obtained by analyzing each term of equation 6.29. In fact, the offset introduced by the first and nth order statistic for classes i and j will cancel each other out, leaving only the average bias between the min and max components of the error (as in equation 6.17), given by $\beta^{spr} = \frac{\beta_i^{1:N} - \beta_j^{1:N} + \beta_i^{N:N} - \beta_j^{N:N}}{s}$.

N	*spread*	*min* or *max*
2	.500	.682
3	.362	.560
4	.299	.492
5	.261	.448
6	.236	.416
7	.219	.392
8	.205	.373
9	.194	.357
10	.186	.344

Table 6.3: Error reduction factors for the spread, *min* and *max* combiners with Gaussian noise model.

The variance of b^{spr} needs to be derived from equation 6.29. Proceeding as in equation 6.18, the variance of the spread combiner can be expressed as:

$$\sigma^2_{b^{spr}} = (\frac{1}{4}\alpha_{1:N} + \frac{1}{4}\alpha_{N:N} + \frac{1}{2}B_{1,N:N})(\sigma^2_{b^m} + \sigma^2_{\beta^m}). \qquad (6.30)$$

For a symmetric distribution (where $\alpha_{1:N} = \alpha_{N:N}$), we obtain the following error:

$$\begin{aligned} E^{spr}_{model}(\beta) &= \frac{s}{2}\mathcal{M}_2 = \frac{s}{2}(\sigma^2_{b^{spr}} + \mathcal{M}_1{}^2) \\ &= \frac{s}{2}\left(\frac{1}{2}\alpha_{1:N} + \frac{1}{2}B_{1,N:N})(\sigma^2_{b^m} + \sigma^2_{\beta^m}) + (\beta^{spr})^2\right) \\ &= \frac{1}{2}(\alpha_{1:N} + B_{1,N:N})E_{model}(\beta) + \\ &\quad \frac{s}{4}(\alpha_{1:N} + B_{1,N:N})(\sigma^2_{\beta^m} - (\beta^m)^2) + \frac{s}{2}(\beta^{spr})^2, \qquad (6.31) \end{aligned}$$

which is very similar to equation 6.21, where the value of α for a single order statistic is now replaced by $\frac{\alpha_{1:N}+B_{1,N:N}}{2}$, since the mean of the first and nth order statistic is used in the posterior estimate.

6.5.2 Trimmed Means

Instead of actively using the extreme values as was the case with the spread combiner, one can base the posterior estimate around the median values. However, instead of selecting one classifier output as was done for f^{med}, one can use multiple classifiers whose outputs are "typical." In this scheme, only a certain fraction of all available classifiers are used for a given pattern. The main advantage of this method over weighted averaging is that the set of classifiers which contribute to the combiner vary from pattern to pattern. Furthermore, they do not need to be determined externally, but are a function of the current pattern and the classifier responses to that pattern.

Trimmed Mean Combiner for Unbiased Classifiers:

Let us formally define the trimmed mean combiner ($\beta_k = 0, \forall k$) as follows:

$$\begin{aligned} f_i^{trim}(x) & \\ &= \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} f_i^{m:N}(x) \\ &= p(c_i|x) + \eta_i^{trim}(x), \end{aligned} \tag{6.32}$$

where:

$$\eta_i^{trim}(x) = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} \eta_i^m(x).$$

The variance of $\eta_i^{trim}(x)$ is given by:

$$\begin{aligned} \sigma^2_{\eta_i^{trim}} & \\ = \frac{1}{(N_2 - N_1 + 1)^2} \sum_{l=N_1}^{N_2} \sum_{m=N_1}^{N_2} cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) & \\ &= \frac{1}{(N_2 - N_1 + 1)^2} \\ \left(\sum_{m=N_1}^{N_2} \sigma^2_{\eta_i^{m:N}(x)} + \sum_{m=N_1}^{N_2} \sum_{l>m}^{N_2} 2cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) \right). \end{aligned} \tag{6.33}$$

Again, using the factors in tables 6.1 and 6.2, equation 6.33 can be further simplified. Note that because the Gaussian distribution is symmetric, the covariance between the kth and lth ordered samples is the same as that between the $N + 1 - k$th and $N + 1 - l$th ordered samples. Therefore, equation 6.33 leads to:

$$\begin{aligned} \sigma^2_{\eta_i^{trim}} &= \frac{1}{(N_2 - N_1 + 1)^2} \sum_{m=N_1}^{N_2} \alpha_{m:N} \sigma^2_{\eta_i(x)} \\ &+ \frac{2}{(N_2 - N_1 + 1)^2} \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \sigma^2_{\eta_i(x)}, \end{aligned} \tag{6.34}$$

N	*ave* (for N)	*trim* (for $N_1 = 2$; $N_2 = N - 1$)	*ave* (for $N - 2$)
3	.333	.449	1.00
4	.250	.298	.500
5	.200	.227	.333
6	.167	.184	.250
7	.143	.155	.200
8	.125	.134	.167
9	.111	.113	.143

Table 6.4: Error Reduction factors for trim and two corresponding *ave* combiners with Gaussian noise model.

where $\alpha_{m:N}$ is the variance of the mth ordered sample and $B_{m,l:N}$ is the covariance between the mth and lth ordered samples, given that the initial samples had unit variance (Sarhan and Greenberg 1956). Using the theory highlighted in section 6.3, and equation 6.34, we obtain the following model error reduction:

$$\frac{E^{trim}_{model}}{E_{model}} = \frac{1}{(N_2 - N_1 + 1)^2} \left(\sum_{m=N_1}^{N_2} \alpha_{m:N} + 2 \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \right). \quad (6.35)$$

Based on equation 6.35 and tables 6.1 and 6.2, we have generated a sample *trim* combiner reduction table. Because there are many possibilities for N_1 and N_2, a table that exhaustively provides all reduction values is not practical. In this sample table we have selected $N_1 = 2$ and $N_2 = N - 1$, that is, averaging after the lowest and highest values have been removed. For comparison purposes the reduction factors of the averaging combiner for N and $N-2$ classifiers are also provided (for i.i.d. classifiers the reduction factors are 1/N as derived in Tumer and Ghosh [1996a]; similar results were obtained for regression problems [Perrone and Cooper 1993]). As these numbers demonstrate, although $N - 2$ classifiers are used in the trim combiner, selectively weeding out undesirable classifiers provides reduction factors significantly better than simply averaging $N - 2$ arbitrary classifiers. The *trim* combiner provides reduction factors comparable the the N classifier *ave* combiner without being susceptible to corruption by one particularly faulty classifier.

Trimmed mean Combiner for Biased Classifiers:

Now, if the classifier biases are nonzero, the trimmed mean combiner's output is given by:

$$\begin{aligned} f_i^{trim}(x) &= \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} f_i^{m:N}(x) \\ &= p(c_i|x) + (\eta_i(x) + \beta_i)^{trim}. \end{aligned} \quad (6.36)$$

In that case the boundary offset is given by:

$$b^{trim} = \frac{(\beta_i + \eta_i(x_b))^{trim} - (\beta_j + \eta_j(x_b))^{trim}}{s}. \qquad (6.37)$$

The first moment of b^{trim} can be obtained from a manner similar to that of the spread combiner. Indeed, each mean offset introduced by a specific order statistic for class i will be offset by the one introduced for class j. Only the trimmed mean of the biases will remain, giving the first moment of b^{trim}:

$$\beta^{trim} = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} \frac{\beta_i^{m:N} - \beta_j^{m:N}}{s}. \qquad (6.38)$$

In deriving the variance of b^{trim}, we follow the same steps as in sections 6.4.3 and 6.5.1. The resulting boundary variance is similar to equation 6.18, but the since the reduction is due to the linear combination of multiple ordered outputs, α is replaced by $\mathcal{A}$, where:

$$\mathcal{A} = \frac{1}{(N_2 - N_1 + 1)^2} \left(\sum_{m=N_1}^{N_2} \alpha_{m:N} + 2 \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \right). \qquad (6.39)$$

The model error reduction in this case is given by:

$$\begin{aligned} E_{model}^{trim}(\beta) &= \frac{s}{2}\mathcal{M}_2 = \frac{s}{2}(\sigma_{b^{trim}}^2 + \mathcal{M}_1{}^2) \\ &= \frac{s}{2}\left(\mathcal{A}(\sigma_{b^m}^2 + \sigma_{\beta^m}^2) + (\beta^{spr})^2\right) \\ &= \mathcal{A}E_{model}(\beta) + \frac{s}{2}(\mathcal{A}(\sigma_{\beta^m}^2 - (\beta^m)^2) + (\beta^{spr})^2). \qquad (6.40) \end{aligned}$$

Once again we need to look at the interaction between the two parts of the error reduction. The first term provides the error reduction compared to the model error of an individual classifier. The smaller $\mathcal{A}$ is, the more error reduction there will be. In the second term, on the other hand, a small value for $\mathcal{A}$ is only useful if the variability in the individual biases is higher than the biases themselves ($\sigma_{\beta^m}^2 > (\beta^m)^2$).

6.6 Experimental Results

The order statistics-based combining methods proposed in this article are tailored for situations where one or more of the following apply: (1) individual classifier performance is uneven and class dependent; (2) it is not possible (insufficient data, high amount of noise) to fine tune the individual classifiers without using computationally expensive methods; and (3) all the features may not be available to all the classifiers.

Such situations occur, for example, in electrical logging while drilling for oil, where data from certain well sites almost completely misses out on portions of the problem space, and in imaging from airborne platforms where the classifiers receive inputs from different satellites and/or different types of sensors (e.g., thermal, optical, SAR). In this article we restrict ourselves to public domain datasets and simulate such variability in two ways, namely, by (1) segmenting the feature set and allowing individual classifiers to have access to only a limited portion of the feature set; and (2) using "early stopping" i.e., prematurely terminating the training of the individual classifiers.[5]

For the experiments reported below, we used a multilayer perceptron (MLP) with a single hidden layer, whose weights were randomly initialized for each run. All classification results reported in this article are test set error rates averaged over 20 runs, along with the differences in the mean (standard deviation divided by square root of the number of runs). Several types of simple combiners such as averaging, weighted averaging, voting, median, products, weighted products (Bayesian), using Dempster-Schafer theory of evidence, and entropy-based averaging, have been proposed in the literature. However, on a wide variety of data sets, it has been observed that simple averaging usually provides results comparable to any of these techniques (and, surprisingly, often better than most of them) (Ghosh, Tumer, Beck, and Deuser 1992; Tumer and Ghosh 1996b). For this reason, in this study, we use the average combiner as a representative of simple combiners, for comparison purposes.

6.6.1 Variability through Segmentation

The first group of experiments focus on classifiers that because of circumstances (e.g., geography) have access to only a part of the full feature set. This situation fits the collective data mining framework wherein a globally distributed dataset is vertically partitioned (see chapter 5). Unfortunately, we are not aware of any public domain datasets for collective data mining, so we instead selected three data sets from the Proben1/UCI benchmarks (Prechelt 1994). Briefly these data sets, and the corresponding size of the MLP used, are[6]: Card: a 51-dimensional, 2-class data set based on credit approval decision (Quinlan 1987), with 690 patterns; an MLP with 10 hidden units; Gene: a 120-dimensional data set with two classes, based on the detection of splice junctions in DNA sequences (Noordewier, Towell, and Shavlik 1991), with

[5] In all the experiments reported here, "high variability" means that classifiers in an ensemble were trained half as along as they would have been, had they been stand-alone classifiers.

[6] The number of hidden units was determined experimentally.

Data	Number of Original Features	Number of Segments	Features per Segment	
			no Overlap	Overlap
Card	51	4	13-13-13-12	18-18-18-18
Gene	120	4	30-30-30-30	40-40-40-40
Sat	36	4	9-9-9-9	15-15-15-15

Table 6.5: Number of features in Proben1/UCI data sets

3175 patterns; an MLP with 10 hidden units; Satellite: a 36-dimensional, 6-class data set with 6435 examples of feature vectors extracted from satellite imagery; an MLP with 20 hidden units.

These three sets were chosen as they have relatively large number of features, somewhat large number of data points, and have been studied by several researchers. Also note that the Proben1 benchmarks are particular training, validation and test splits of the UCI data sets.[7] The results presented in this article are based on the first training, validation and test partition discussed in Prechelt (1994), where half the data is used for training, and a quarter each for validation and testing purposes.

We investigate two situations: one where the original features were randomly and disjointly partitioned among the different segments, and the second where there is some overlap among features in different segments. The exact segment count and number of features within each segment is specified in table 6.5.

For each data set, we present the original number of features, the number of new features sets that result when the feature set is segmented (for Gene we only have two new sets, because the low dimensionality prevents any further segmentation), and the resulting number of features in each segment with and without overlap among the features.

A classifier trains on data from one segment, and different classifiers operate on different segments. When the number of combiners were higher than the number of segments ($N = 8$) more than on classifier (starting from a different initialization) was trained on the same features.

Tables 6.6-6.7 present the results (with the best result for each case in bold font). The misclassification percentage for individual classifiers are reported in the first column. For the trimmed mean combiner, we also provide N_1 and N_2, the upper and lower cutting points in the ordered average used in equation 6.32, obtained through the validation set.

In this case, for two of the three data sets (Gene and Card), there are striking gains due to using order statistics combiners. One cause for these gains is the high variability in performance among the component classifiers. In such cases, a small number of poor classifiers can corrupt the average combiner. By

[7] available from www.ics.uci.edu/~mlearn/MLRepository.html

Data	N	Ave	Max	Min	Spread	Trim (N_1-N_2)
Card	4	12.21 ± .00	**10.58 ± .06**	**10.61 ± .06**	**10.58 ± .00**	12.21 ± .00 (3-4)
30.30 ± 2.62	8	12.21 ± .00	**10.47 ± .00**	10.61 ± .06	**10.47 ± .00**	**10.47 ± .00** (7-8)
Gene	4	18.52 ± .10	**14.02 ± .13**	20.23 ± .31	14.72 ± .15	16.86 ± .15 (3-4)
34.80 ± 4.01	8	18.06 ± .06	**13.13 ± .06**	17.59 ± .17	13.69 ± .11	13.39 ± .08 (7-8)
Sat	4	14.16 ± .08	14.73 ± .18	14.64 ± .16	14.24 ± .12	**14.00 ± .07** (3-4)
16.40 ± 0.56	8	14.21 ± .05	15.27 ± .15	15.07 ± .15	14.49 ± .11	**14.01 ± .04** (3-5)

Table 6.6: Segmented features with overlap (% misclassified $\pm\sigma/\sqrt{n}$).

Data	N	Ave	Max	Min	Spread	Trim (N_1-N_2)
Card	4	12.21 ± .00	**10.49 ± .03**	**10.49 ± .03**	**10.49 ± .03**	12.21 ± .00 (3-4)
30.90 ± 2.66	8	12.21 ± .00	10.78 ± .07	10.78 ± .07	**10.52 ± .04**	11.05 ± .00 (7-8)
Gene	4	24.35 ± .13	15.82 ± .15	19.15 ± .22	**14.09 ± .11**	23.11 ± .15 (3-4)
36.87 ± 3.01	8	23.33 ± .19	14.99 ± .14	16.78 ± .24	**13.23 ± .15**	15.03 ± .17 (7-8)
Sat lap	4	14.39 ± .09	15.66 ± .15	15.46 ± .11	15.11 ± .11	**14.22 ± .07** (2-3)
17.13 ± 0.47	8	14.37 ± .05	15.93 ± .06	15.53 ± .06	15.18 ± .10	**14.04 ± .05** (3-5)

Table 6.7: Segmented features without overlap (% misclassified $\pm\sigma/\sqrt{n}$).

their very nature, though, combiners based on order statistics are immune to this type of corruption. The ave combiner performs well on the Sat data sets where the performance among the individual classifiers is much more homogeneous. In this case, the ave results are only marginally worse than those for the trimmed mean.

6.6.2 Variability through Early Stopping

For the second set of experiments we use two classes of acoustic underwater sonar signals[8]. From the original sonar signals of four different underwater objects (porpoise sound, cracking ice and two different whale sounds), two feature sets are extracted (Ghosh, Deuser, and Beck 1992): (1) WOC: a 25-dimensional feature set, consisting of Gabor wavelet coefficients, temporal descriptors and spectral measurements; and, (2) RDO: a 24-dimensional feature set, consisting of reflection coefficients based on both short and long time windows, and temporal descriptors. For both feature sets, an MLP with 50 hidden units was used.[9] Further details about this 4-class problem can be found in Ghosh, Deuser, and Beck (1992) and Tumer and Ghosh (1996b).

Table 6.8 presents the combining results for the underwater acoustic data set when the individual classifier performance is highly variable. The results of table 6.8 as well as those given in Tumer and Ghosh (1999) indicate that when the individual classifier performance is highly variable, order statistics-based

[8] Results on 6 Proben/UCI datasets were reported in Tumer and Ghosh (1999) and hence are not repeated here

[9] These data sets are available at www.lans.ece.utexas.edu.

Data	N	Ave	Max	Min	Spread	Trim (N_1-N_2)
RDO	4	11.57 ± .11	11.94 ± .12	11.52 ± .20	**11.04 ± .09**	11.34 ± .14 (3-4)
13.32 ± 0.83	8	11.64 ± .09	11.47 ± .11	**11.29 ± .13**	11.51 ± .09	12.30 ± .08 (4-5)
WOC	4	8.80 ± .09	**7.84 ± .10**	9.31 ± .12	8.54 ± .06	8.43 ± .13 (3-4)
12.07 ± 1.12	8	8.82 ± .08	**7.68 ± .12**	8.91 ± .06	8.24 ± .11	**7.81 ± .08** (7-8)

Table 6.8: Combining results in the presence of high variability in individual classifier performance for the sonar data (% misclassified $\pm\sigma/\sqrt{n}$).

combiners (particularly the *spread* combiner) typically provide better classification results than other simple combiners. This performance improvement is obtained without sacrificing the simplicity of the combiner. One important thing to note, however, is that in all cases studied, the order statistics based combiners performed at least as well as the simple combiner, implying that no risk is taken by using this method.

A close inspection of these results reveals that using either the max or min combiner can provide better classification rates than *ave*, but it is difficult to determine which of the two will be more successful given a data set. A validation set may be used to select one over the other, but in that case, potentially precious training data is used solely for determining which combiner to use. The use of the spread combiner removes this dilemma by consistently providing results that are comparable to, or better than, the best of the max-min duo.

6.7 Concluding Remarks

In this article we present and analyze combiners based on order statistics. These combiners blend the simplicity of averaging with the generality of metalearners. They are particularly effective if there are significant variations among component classifiers in at least some parts of the joint input-output space. Variations can arise when the individual training sets cannot be considered as random samples from a common universal data set. Examples of such cases include real-time data acquisition and classification from geographically distributed sources or data mining problems with large and possibly heterogenous databases, where random subsampling is computationally expensive and practical methods lead to nonrandom subsamples (Bradley and Fayyad, 1998). The robustness of order statistics combiners is also helpful when certain individual classifiers experience catastrophic failures (e.g., due to faulty sensors).

The analytical framework provided in this chapter quantifies the reductions in error achieved when an order statistics based ensemble is used. It also shows that the two methods for linear combination of order statistics introduced in this chapter provide more reliable estimates of the true posteriors than any of the individual order statistic combiners.

The experimental results of section 6.5 indicate that when there is significant variability among the classifiers, the order statistics-based combiners substantially outperform simple combiners. Our previous results also showed that in the absence of such variability these combiners perform no worse. Thus the family of order statistic combiners are applicable over a wide range of situations. They are able to extract an appropriate amount of information from the individual classifier outputs without requiring tuning additional parameters as in metalearners, and without being substantially affected by outliers.

A future endeavor, which will be helpful for this work as well as for the study of collective data mining on very large datasets in general, is to obtain a suite of public domain datasets which are intrinsically partitioned into segments with varying quality. Though such situations sometimes occur in practice (for example in oil logging data (Chakravarthy 1997) and mortgage scoring (Merz 1998); both data sets proprietary), they are not represented in the standard, venerable databases such as UCI, ELENA and Statlog typically used by the academic community. Perhaps the recent CRoss-Industry Standard Process for Data Mining (CRISP-DM) initiative will provide a satisfactory solution to this problem in the near future.

Acknowledgments

This research was supported in part by ARO contracts DAAG55-98-1-0230 and DAAD19-99-1-0012, and NSF grant ECS-9900353.
AUTHOR: AAAI PRESS FOLLOWS A "DOWN" STYLE FOR PARTS OF THE BOOK AND MOST TITLES (UNLESS A PROPER NAME). (figure, table, chapter, etc.). ALSO NOTE THAT CHICAGO MANUAL OF STYLE AND AAAI PRESS STYLE CALL FOR ALMOST ALL PREFIXES TO BE SPELLED SOLID (NO HYPHEN) EVEN WHEN USED AS ADJECTIVES. ALSO, MOST LISTS HAVE BEEN CONVERTED TO PARAGRAPH STYLE TO CONSERVE SPACE (BOOK IS TOO LONG!).